**Проверочная работа 7**

1. Какой из запросов будет соответствовать сортировке такого вывода?

|  |  |
| --- | --- |
| **col\_1** | **col\_2** |
| 1 | C |
| 1 | B |
| 1 | A |
| 2 | C |
| 2 | B |
| 2 | A |
| 3 | C |
| 3 | B |
| 3 | A |

Вариант №1:

|  |  |
| --- | --- |
|  | SELECT \*  FROM Table\_1  ORDER BY col\_1 DESC, col\_2 ASC |

Вариант №2:

|  |  |
| --- | --- |
|  | SELECT \*  FROM Table\_1  ORDER BY col\_1 ASC, col\_2 DESC |

Вариант №3:

|  |  |
| --- | --- |
|  | SELECT \*  FROM Table\_1  ORDER BY col\_1, col\_2 ASC |

1. Что будет в результате выполнения следующего скрипта?

BEGIN TRAN

TRUNCATE TABLE Employees

ROLLBACK

SELECT \* FROM Employees

Выведет все, что есть в таблице Employees, т.к. ROLLBACK

- это мы откатили свои изменения до BEGIN, т.е. удаление таблицы Employees не произошло.

3. Какой вывод соответствует запросу ниже с объединением JOIN?

В БД имеются 2 таблицы:

|  |
| --- |
| Table\_1 |
| **col\_1** |
| 1 |
| 2 |
| 4 |
| NULL |
| NULL |

|  |
| --- |
| Table\_2 |
| **col\_1** |
| 1 |
| 2 |
| 3 |
| NULL |

|  |  |
| --- | --- |
|  | SELECT \*  FROM Table\_1  JOIN Table\_2      ON Table\_1.col\_1 = Table\_2.col\_1 |

Какой вывод вернет база?

Значения null будут игнорироваться, т.к. мы не можем null ни с чем сравнить, в том числе с другим null

|  |  |
| --- | --- |
| co11 | col1 |
| 1 | 1 |
| 2 | 2 |

1. Какой вывод соответствует запросу ниже с объединением LEFT JOIN?

В БД имеются 2 таблицы:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Table\_1 | | **col\_1** | | 1 | | 2 | | 4 | | NULL | | NULL | | |  | | --- | | Table\_2 | | **col\_1** | | 1 | | 2 | | 3 | | NULL | |

К БД посылается такой запрос:

|  |  |
| --- | --- |
|  | SELECT \*  FROM Table\_1  LEFT JOIN Table\_2      ON Table\_1.col\_1 = Table\_2.col\_1 |

Какой вывод вернет база?

Значения null из правой таблицы будут игнорироваться, а из левой таблицы мы выведем все null

|  |  |
| --- | --- |
| co11 | col1 |
| 1 | 1 |
| 2 | 2 |
| 4 | null |
| null | null |
| null | null |

5. Какое значение будет возвращено в результате следующего запроса?

SELECT AVG(TotalPayment)

FROM Orders

|  |  |
| --- | --- |
| OrderNo | TotalPayment |
|  |  |
| 1 | 100 |
|  |  |
| 2 | 200 |
|  |  |
| 3 | 100 |
|  |  |
| 4 | NULL |
|  |  |
| 5 | 100 |
|  |  |
| 6 | 100 |
|  |  |

120

1. Чем различаются запросы ниже?

Представленные запросы выполняются к одной и той же таблице одинакового наполнения. Таблица состоит из одного столбца Col\_1.

Запрос №1:

|  |  |
| --- | --- |
|  | SELECT COUNT(\*) AS Cnt  FROM Table\_1 |

Запрос №2:

|  |  |
| --- | --- |
|  | SELECT COUNT(DISTINCT Col\_1) AS Cnt  FROM Table\_1 |

Первый запрос пробежится по всей таблице и посчитает количество строк всего в таблице.

Второй запрос вначале «удалит» все повторяющиеся значения из поля «Col\_1», а затем посчитает количество этих уникальных значений (null будет пропущен).

1. Какой тип отношения между таблицами «Покупатели» и «Заказы»?

Имеется база данных интернет-магазина. В этой БД имеются таблицы:

* Покупатели – хранит информацию по клиентам (ФИО, логин, пароль и т.п.);
* Заказы – содержит номер заказа, комментарий, дату доставки, покупателя (который сделал заказ) и т.д.;
* Товары – id товара, наименование, цена, остатки и т.п.;

    Один к одному (в табллице «Покупатели» один покупатель, который может встречаться много раз в таблице «Заказы»)

    Один ко многим

    Многие ко многим

    Таблицы никак не связаны

1. Какой тип отношения между таблицами «Товары» и «Заказы»?

Имеется база данных интернет-магазина. В этой БД имеются таблицы:

* Покупатели – хранит информацию по клиентам (ФИО, логин, пароль и т.п.);
* Заказы – содержит номер заказа, комментарий, дату доставки, покупателя (который сделал заказ) и т.д.;
* Товары – id товара, наименование, цена, остатки и т.п.;

    Один к одному

    Один ко многим (здесь я полагаю, что в описании таблицы «Заказы» под «и т.д.» подразумевается, естественно, и наименование товара. Но если описание полей таблицы «Заказы полное и под «и т.д. ничего не храниться, то связи между таблицами нет)

    Многие ко многим

    Таблицы никак не связаны

1. В какой нормальной форме находится база данных?

В какой нормальной форме находится база данных?  
Имеется БД из одной таблицы «Сотрудники».  
Таблица содержит столбцы:

* Id – уникальный идентификатор сотрудника;
* ФИО – содержит фамилию, имя и отчество сотрудника;
* Дата рождения – дата рождения пользователя;
* Отдел\_id – идентификатор отдела, в котором пользователь работает;
* Отдел – наименование отдела, в котором пользователь работает.

    БД в первой нормальной форме (предполагаю, что в одной ячейке содержится одно значение)

    БД во второй нормальной форме (не подходит, т.к. отдел зависит не от первичного ключа Id, а от поля Отдел\_id)

    БД в третьей нормальной форме

    Ни одна нормальная форма не соблюдена

1. Какие строки вернет запрос c LIKE?

Имеется таблица:

|  |  |
| --- | --- |
| **ID** | **Value** |
| 1 | AAA |
| 2 | BBB |
| 3 | CCC |
| 4 | ABC |
| 5 | CBA |
| 6 | 123 |
| 7 | 213 |
| 8 | 321 |
| 9 | 1AB |
| 10 | BC3 |

К таблице выполняется запрос:

|  |  |
| --- | --- |
|  | SELECT \*  FROM Table\_1  WHERE id like '%A\_' |

Какие строки будут возвращены?

Никаких значений не вернет, т.к. в id нет букв вообще

1. Даны следующие таблицы:





Что будет результатом для следующего скрипта?

SELECT \* FROM runners

WHERE id NOT IN (SELECT winner\_id FROM races)

Ничего не вернет, т.к. подзапрос вернет (2, 3 , null), а на прошлом уроке мы прорешали на примерах, что если IN, то запрос будет возвращать значения, игнорирую null; а вот если NOT IN, то уже ничего не вернет (даже если кроме null будут соответствия).

1. Даны следующие данные в колонке:

|  |
| --- |
| **x** |
| -2 |
| 2 |
| -5 |
| 0 |
| 1 |

Напишите скрипт для нахождения суммы всех положительных значений из колонки x или значений, кратных 5.

SELECT SUM(x)

FROM table1

WHERE x%5 = 0 OR x > 0